

Requested Patent: JP11120113A

Title:

SYSTEM FOR SELECTIVELY UPGRADING FIRMWARE CODE FOR OPTICAL DISK  
DRIVE VIA ATA/IDE INTERFACE BASED ON HOST SYSTEM PROGRAMMING  
ENABLE SIGNAL ;

Abstracted Patent: US5968141 ;

Publication Date: 1999-10-19 ;

Inventor(s): TSAI HSI-JUNG (TW) ;

Applicant(s): WINBOND ELECTRONICS CORP (TW) ;

Application Number: US19970904541 19970804 ;

Priority Number(s): TW19970108241 19970614 ;

IPC Classification: G06F13/10 ;

Equivalents: ;

**ABSTRACT:**

An apparatus and the method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface are disclosed. The optical disk drive has a firmware code memory and a microcontroller that executes the firmware code. A drive decoder decodes to connect the optical disk drive to a host computer system via the ATA/IDE interface. The apparatus includes a programming controller that receives signals from the ATA/IDE interface and performs input/output decode to determine whether the host computer system has requested an upgrade of the firmware or to maintain the optical disk drive in normal operation. A multiplexer has a first input connected to the programming controller, and the second input connected to the microcontroller. The multiplexer selects the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests a firmware upgrade. The on-site upgrade operation can be performed without access to the inside of the drive unit.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-120113

(43) 公開日 平成11年(1999) 4月30日

(51) Int.Cl.<sup>6</sup>  
 G 0 6 F 13/10  
 3/08  
 G 1 1 B 20/10

識別記号  
 3 2 0

F I  
 G 0 6 F 13/10  
 3/08  
 G 1 1 B 20/10

3 2 0 A  
 F  
 D

審査請求 有 請求項の数29 OL 外国語出願 (全 49 頁)

(21) 出願番号 特願平9-238562

(22) 出願日 平成9年(1997) 9月3日

(31) 優先権主張番号 8 6 1 0 8 2 4 1

(32) 優先日 1997年6月14日

(33) 優先権主張国 台湾 (TW)

(71) 出願人 595039162

華邦電子股▲ふん▼有限公司

台湾新竹科學園區研新三路4號

(72) 発明者 蔡 錫榮

台湾新竹縣寶山▲郷▼雙溪村雙豐路144號

6樓

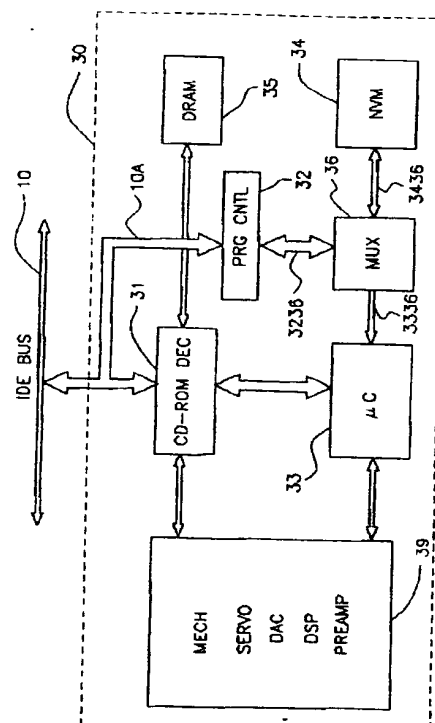
(74) 代理人 弁理士 秋元 輝雄

(54) 【発明の名称】 光ディスクドライブにおけるファームウェア コードをアップグレードする方法

(57) 【要約】 (修正有)

【課題】 ATA/IDEインターフェースを介して光学ディスクドライブのファームウェアコードをアップグレードする。

【解決手段】 本装置は、ATA/IDEインターフェースから信号を受信するプログラミングコントローラ32を有し、入力/出力デコードを実行してホストコンピュータシステムがファームウェアのアップグレードを要求しているか、光学ディスクドライブが通常動作を維持することを要求しているかを決定する。多重化装置36は、プログラミングコントローラに接続される第1の入力と、マイクロコントローラに接続される第2の入力とを有する。多重化装置は、ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、アップグレード動作を実行するために、第1の入力を多重化出力を介してメモリデバイスに切替える。ドライブユニットの内部にアクセスすることなく、オンサイトのアップグレード動作が実行できる。



## 【特許請求の範囲】

【請求項1】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする装置であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラと、前記ATA/IDEインターフェースを介して前記光学ディスクドライブをホストコンピュータシステムに接続するためのデコードをするドライブデコードとを備える装置において、

前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するプログラミングイネーブル信号を生成するプログラミングコントローラ手段と、

第1及び第2の入力、並びに多重化出力及び多重化選択入力を有し、前記第1の入力は前記プログラミングコントローラ手段に接続され、前記第2の入力は前記マイクロコントローラに接続され、そして前記多重化選択入力は前記プログラミングイネーブル信号を受信する多重化手段とを備え、

前記多重化手段は、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したときに、ファームウェアアップグレード動作を行うために、前記多重化出力を介して前記第1の入力を前記メモリデバイスに切換えることを特徴とする装置。

【請求項2】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項1の装置において、前記プログラミングコントローラ手段が更に、

前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行ってプログラミングイネーブル信号を生成し、更に第1、第2、第3及び第4のイネーブル信号を生成する入力/出力デコードと、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信してアップグレード初期化キー信号を生成する初期化キーイネーブルロジックと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第2のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みアドレス信号を生成するアドレスプログラムイネーブル

ラッチと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第3のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みデータ信号を生成するデータプログラムイネーブルラッチと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第4のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込み制御信号を生成する制御プログラムイネーブルラッチとを備えることを特徴とする装置。

【請求項3】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項2の装置において、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アドレスプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースの合計16ビットのアドレス信号であることを特徴とする装置。

【請求項4】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項2の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記データプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項5】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項2の装置において、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記制御プログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項6】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項2の装置において、前記光学ディスクドライブは、CD-ROMドライブであることを特徴とする装置。

【請求項7】 ATA/IDEインターフェースを介し

て光学ディスクドライブ用のファームウェアコードをアップグレードする請求項2の装置において、前記光学ディスクドライブは、DVDドライブであることを特徴とする装置。

【請求項8】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする装置であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラと、前記ATA/IDEインターフェースを介して前記光学ディスクドライブをホストコンピュータシステムに接続するためのデコードをするドライブデコーダとを備える装置において、

前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するプログラミングイネーブル信号を生成するプログラミングコントローラ手段と、

第1及び第2の入力と、多重化出力及び多重化選択入力とを有し、前記第1の入力は前記プログラミングコントローラ手段に接続され、前記第2の入力は前記マイクロコントローラに接続され、そして前記多重化選択入力は前記プログラミングイネーブル信号を受信する多重化手段とを備え、

前記プログラミングコントローラ手段が、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行ってプログラミングイネーブル信号を生成し、更に第1、第2、第3及び第4のイネーブル信号を生成する入力/出力デコーダと、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信してアップグレード初期化キー信号を生成する初期化キーイネーブルロジックと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第2のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みアドレス信号を生成するアドレスプログラムイネーブルラッチと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第3のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信

し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みデータ信号を生成するデータプログラムイネーブルラッチと、

前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第4のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込み制御信号を生成する制御プログラムイネーブルラッチとを備えることを特徴とする装置。

【請求項9】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項8の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アドレスプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースの合計16ビットのアドレス信号であることを特徴とする装置。

【請求項10】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項8の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記データプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項11】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項8の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記制御プログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項12】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項8の装置において、前記光学ディスクドライブは、CD-ROMドライブであることを特徴とする装置。

【請求項13】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項8の装置において、前記光学ディスクドライブは、DVDドライブであることを特徴とする装置。

【請求項14】 ATA/IDEインターフェースを介

して光学ディスクドライブ用のファームウェアコードをアップグレードする方法であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラとを備える方法において、

前記光学ディスクドライブが、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って、前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するステップと、  
前記光学ディスクドライブが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したときに、前記ATA/IDEインターフェースを前記メモリデバイスに接続し、前記ホストコンピュータが前記メモリデバイスに直接書き込むことによるファームウェアアップグレード動作の実行を許容し、そして前記ホストコンピュータシステムが前記光学ディスクドライブの通常動作を要求したときに、前記マイクロコントローラを前記メモリデバイスに接続するステップとを備えることを特徴とする方法。

【請求項15】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項14の方法において、前記光学ディスクドライブは、CD-ROMドライブであることを特徴とする方法。

【請求項16】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項14の方法において、前記光学ディスクドライブは、DVDドライブであることを特徴とする方法。

【請求項17】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする方法であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラと、前記ATA/IDEインターフェースを介して前記光学ディスクドライブをホストコンピュータシステムに接続するためのデコードをするドライブデコードと、第1及び第2の入力、並びに多重化出力及び多重化選択入力を有し、前記第1の入力はプログラミングコントローラ手段に接続され、前記第2の入力は前記マイクロコントローラに接続される多重化手段と、前記多重化手段の多重化動作を選択的に切替える前記プログラミングコントローラ手段とを備える方法において、  
前記プログラミングコントローラ手段が、前記ATA/

IDEインターフェースから信号を受信し、そして入力/出力デコードを行って前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するプログラミングイネーブル信号を生成するステップと、  
多重化選択入力を有する前記多重化手段が、前記プログラミングイネーブル信号を受信し、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1の入力を前記多重化出力を介して前記メモリデバイスに切替えるステップとを備えることを特徴とする方法。

【請求項18】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項17の方法において、前記プログラミングコントローラ手段が前記プログラミングイネーブル信号を生成するステップが更に、  
前記プログラミングコントローラ手段の入力/出力デコードが、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行ってプログラミングイネーブル信号を生成し、更に第1、第2、第3及び第4のイネーブル信号を生成するステップと、  
前記プログラミングコントローラ手段の初期化キーイネーブルロジックが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信してアップグレード初期化キー信号を生成するステップと、前記プログラミングコントローラ手段のアドレスプログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第2のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みアドレス信号を生成するステップと、  
前記プログラミングコントローラ手段のデータプログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第3のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みデータ信号を生成するステップと、  
前記プログラミングコントローラ手段の制御プログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第4のイネーブル信号と前記ATA/IDEインターフェースからの対

応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込み制御信号を生成するステップとを備えることを特徴とする方法。

【請求項19】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項18の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アドレスプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースの合計16ビットのアドレス信号であることを特徴とする方法。

【請求項20】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項18の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記データプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項21】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項18の方法において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記制御プログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする方法。

【請求項22】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項18の方法において、前記光学ディスクドライブは、CD-ROMドライブであることを特徴とする方法。

【請求項23】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項18の方法において、前記光学ディスクドライブは、DVDドライブであることを特徴とする方法。

【請求項24】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする方法であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラと、前記ATA/IDEインターフェースを介して前記光学ディスクドライブをホストコンピュータシステムに接続するための

デコードをするドライブデコーダと、第1及び第2の入力、並びに多重化出力及び多重化選択入力とを有し、前記第1の入力はプログラミングコントローラ手段に接続され、前記第2の入力は前記マイクロコントローラに接続される多重化手段と、前記多重化手段の多重化動作を選択的に切替える前記プログラミングコントローラ手段とを備える方法において、

前記プログラミングコントローラ手段が、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するプログラミングイネーブル信号を生成するステップと、

多重化選択入力とを有する前記多重化手段が、前記プログラミングイネーブル信号を受信し、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1の入力を前記多重化出力を介して前記メモリデバイスに切替えるステップとを備え、

前記プログラミングイネーブル信号を生成するステップが更に、

前記プログラミングコントローラ手段の入力/出力デコーダが、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行ってプログラミングイネーブル信号を生成し、更に第1、第2、第3及び第4のイネーブル信号を生成するステップと、

前記プログラミングコントローラ手段の初期化キーイネーブルロジックが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記第1のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信してアップグレード初期化キー信号を生成するステップと、

前記プログラミングコントローラ手段のアドレスプログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第2のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みアドレス信号を生成するステップと、

前記プログラミングコントローラ手段のデータプログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第3のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込みデータ信号を生成するステップと、

前記プログラミングコントローラ手段の制御プログラムイネーブルラッチが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アップグレード初期化キー信号と前記第4のイネーブル信号と前記ATA/IDEインターフェースからの対応する信号とを受信し、前記ATA/IDEインターフェース上の前記対応する信号をラッチして、前記メモリデバイス用の書き込み制御信号を生成するステップとを備えることを特徴とする方法。

【請求項25】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項24の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記アドレスプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースの合計16ビットのアドレス信号であることを特徴とする方法。

【請求項26】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項24の装置において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記データプログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする装置。

【請求項27】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項24の方法において、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したとき、前記制御プログラムイネーブルラッチによって受信された前記ATA/IDEインターフェースからの対応する信号は、前記ATA/IDEインターフェースのデータ信号の少なくとも下位8ビットであることを特徴とする方法。

【請求項28】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項24の方法において、前記光学ディスクドライブは、CD-ROMドライブであることを特徴とする方法。

【請求項29】 ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする請求項24の方法において、前記光学ディスクドライブは、DVDドライブであることを特徴とする方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的に光学ディスクドライブのファームウェアコードをオンサイト (on

-site:実装状態)でアップグレード(改良)することに関する。特に、本発明は、光学ディスクドライブ用の装置と、それに対応してホストコンピュータシステムのATA/IDEインターフェースを経由したファームウェアコードのアップグレードを可能にする方法に関する。

【0002】

【従来の技術】光学ディスクドライブ、特にCD-ROMドライブの部類は、光電式デジタルメモリ記憶装置として一般的になり、ほとんどのパーソナルコンピュータシステムに少なくとも1台が装着されるようになっていく。パーソナルコンピュータの世界では、CD-ROMドライブの設計は、最初からCD-ROMディスク上の多量な情報の記憶用に意図されている。この考えは、ある意味では、記憶媒体の表面に情報を記憶する磁気ディスクドライブと基本的に同様である。この種のCDファミリーのメンバーは、ユーザが書き込みできないため、"CD-ROM"と呼ばれる。しかしながら、安価に量産されるCD-ROMからは多量のデータがアクセスできる。

【0003】CD-ROMは、ISOによってデジタルデータ記憶の基準に採用されているが、多量なデータの単純な記憶装置としての他の応用もある。適切なソフトウェアの制御の下で、CD-ROMドライブは、CDファミリーの複数のタイプのディスクのアクセスにも使用できる。

【0004】例えば、パーソナルコンピュータシステムでは、CD-ROMドライブはしばしば従来のCDプレイヤーとして、正に第1世代のCD、即ち音楽CDに記録されている音楽を再生することに使用される。初期の音楽CDの導入の後に、ファミリーにビデオCD(VCD)のようなフォーマットが出現している。従来のパーソナルコンピュータ用CD-ROMドライブは、VCDビデオプログラムをコンピュータの表示画面上に再生できるように設計されている。MPEG-1に採用された固有の制限によって、VCD基準は重大な適用のためではなく、そして2つの電気機器企業グループが最近デジタルビデオディスク(又はデジタルバーサタイルディスク)として知られる新たな基準を達成した。

【0005】これは新たなデータ記憶のフォーマットであり、高解像度の使用による実質的に良質のビデオ効果を提供するMPEG-2基準に基づくCDファミリーの新たなメンバーである。DVD(デジタルビデオディスク)基準が提案されているため、CD-ROMフォーマットに対する新世代のDVDドライブの新たなシステムでは、古いタイプに対する互換性が設計されている。換言すれば、これから発売されるDVDドライブは、そのデフォルトDVDに加えて、CD-ROMを読み取ることが可能である。

【0006】かくして、コンピュータシステムの光学コンパクトディスク(CD)サブシステムとして、CD-

ROMドライブは、媒体のCDファミリーの異なるフォーマットをアクセスできるようにプログラムできる。これは、周知のx86ベースのIBM互換機では、ソフトウェアドライバインストールとして知られる手法によって実行される。オペレーティングシステムによって管理されるコンピュータシステムを、適切なソフトウェアドライバをインストールすることによって、CD-ROM、プレイバックCDまたはビデオCDからのデータファイル読み出しに使用できる。これらのソフトウェアドライバは、基本的にCD-ROMドライブの制御システムにおけるハードウェアレベルの制御プログラム領域の最上部に構築されている。このファームウェアプログラムは通常不揮発性半導体メモリデバイス、例えば消去可能なプログラマブル読み出し専用メモリ(EPROM)や、電氣的に消去可能なプログラマブル読み出し専用メモリ(EEPROM)内にストアされる。

【0007】このファームウェアに基づき、CD-ROMドライブはホストコンピュータシステムから制御可能になり、適切に音楽CDプレイヤ、VCDプレイヤ、及び/又はCD-ROMデータドライブの機能を実行できる。しかしながら、ある種の環境下では、特にCD-ROMドライブの新たなモデルが市場で販売されると、ソフトウェアプログラムは、ドライブそのものの制御電子回路内のファームウェアであれ、オペレーティングシステムレベル/アプリケーションレベルのドライブであれ、ドライブに機能不全を引き起こすプログラム上の問題(プログラムバグ)を持つことがある。この問題がオペレーティングシステムレベル/アプリケーションレベルのドライブに生じた場合は、その修理にはCD-ROMドライブを何等巻き込まずに済む。しかしながら、不幸にして前記の問題がCD-ROMドライブのファームウェアに生じた場合は、ソフトウェアの修理は不可避免的に多くのトラブルを引き起こす。

【0008】基本的に、ドライブファームウェアが何らかのプログラム上の問題を有している場合は、半導体メモリ内にストアされたファームウェアはアップグレード(upgrade)、即ち問題の無い他のコピーによって置き換えられなければならない。ファームウェアプログラムの保持に使用された半導体メモリがEPROMのようなデバイスであれば、その記憶内容を再プログラミングを実行するために、それは挿入ソケットから取り外され、あるいは半田付けされた印刷回路基板から取り外される。通常、これにはEPROMの記憶内容を消去するための紫外線照射が含まれる。通常その様であるが、消去されたメモリデバイスが再使用される場合、その後新たな問題の無いファームウェアプログラムを保持する再プログラミングが必要になる。EEPROMがドライブファームウェアのストアに使用される場合でも、EPROMと同様の手法で取り扱われる必要がある。これは、従来のCD-ROMドライブ制御ロジックが、オンサイト

の再プログラミング用の設計を欠いているためである。

【0009】従来のCD-ROMドライブに対するファームウェアの上述したアップグレード/修理の処置は、ファームウェアメモリデバイスに接近できるように、ドライブのケースが開放可能であることを要求する。メモリデバイスに接近する前段階として、CD-ROMドライブそのものが、既に装着されているのであれば、コンピュータシステムユニットのドライブ装着部から取り外される必要がある。更に、EPROM及び/又はEEPROMデバイスの消去とプログラミングは通常専用の装置、例えば紫外線消去装置及びプログラミング装置を必要とする。それ故、CD-ROMドライブに対する典型的なファームウェアアップグレード動作は、一般的にコンピュータシステムのエンドユーザ側では行われぬ。より好ましいシナリオは、サービス要員にシステムからCD-ROMドライブを取り外させ、更にそのドライブを、ファームウェアをアップグレードできる製造元に返送させることである。

【0010】ファームウェアがアップグレードされるべきCD-ROMドライブが製造元の施設に返送されると、ケースが開放され、メモリデバイスがドライブの電子回路から適当な治具や装置を使用して取り外される。再プログラミングされると、正しいファームウェアプログラムコードを含むメモリデバイス、またはその代用品がドライブの電子回路に装着し直される。十分な試験処理の後に、アップグレードされたCD-ROMドライブは所有者の下に返却されることになる。再び、資格のある要員がそのドライブをコンピュータシステムに組み込む必要がある。明らかなように、CD-ROMのアップグレードに必要な全ての仕事を遂行するためには、比較的に煩雑で専門的な知識が必要とされる。サービス競争がCD-ROMドライブの価格を、製造者がそれまでの利益マージンを維持する事が困難になるまで低下させるとき、その様なアップグレード処理は確実に大きな損害のもとになる。

【0011】これは、製品の保証期間内に何回もCD-ROMドライブに対するファームウェアのアップグレードを実行する必要性が生ずるからである。その様なとき、製造者は、アップグレードを遂行するに必要な全ての追加的な人件費に付いては言うに及ばず、輸送に要する費用と同様に、ホストコンピュータシステムからドライブを取り外す費用を負担する責任が生ずる。一方、アップグレードの工程では、ドライブとその部品が容易に損傷して追加的な損失を発生させることがある。ドライブの所有者にとっては、コンピュータシステムからドライブを取り外すための技術者を探し出す費用が、しばしば安価なCD-ROMドライブの価格のレベルと同じになることがある。

【0012】

【発明が解決しようとする課題】それ故、本発明の目的



は、ドライブユニットをコンピュータシステムから取り外したり、キャビネットを開放する必要がなく、ホストコンピュータシステムのATA/IDEインターフェースを介して光学ディスクドライブにファームウェアコードをオンサイトでアップグレードするための装置とそれに対応する方法を提供することにある。

【0013】本発明の他の目的は、ファームウェアコードを保持するメモリにホストプロセッサが直接書き込みを行うことにより、ホストコンピュータシステムのATA/IDEインターフェースを介して光学ディスクドライブにファームウェアコードをオンサイトでアップグレードするための装置とそれに対応する方法を提供することにある。

【0014】

【課題を解決するための手段】上記目的を達成するため、本発明では、ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする装置であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラと、前記ATA/IDEインターフェースを介して前記光学ディスクドライブをホストコンピュータシステムに接続するためのデコードをするドライブデコーダとを備える装置を提供する。この装置は、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定するプログラミングイネーブル信号を生成するプログラミングコントローラを備える。マルチプレクサは、第1及び第2の入力、並びに多重化出力及び多重化選択入力とを有し、前記第1の入力は前記プログラミングコントローラ手段に接続され、前記第2の入力は前記マイクロコントローラに接続され、そして前記多重化選択入力は前記プログラミングイネーブル信号を受信する。前記マルチプレクサは、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したときに、ファームウェアアップグレード動作を行うために、前記多重化出力を介して前記第1の入力を前記メモリデバイスに切換える。

【0015】本発明の上記他の目的を達成するため、本発明は更に、ATA/IDEインターフェースを介して光学ディスクドライブ用のファームウェアコードをアップグレードする方法であって、前記光学ディスクドライブは、前記ファームウェアコードをストアするためのメモリデバイスと、前記ファームウェアコードを実行して前記ドライブによる前記光学ディスク内容のアクセス動作を制御するマイクロコントローラとを備える方法を提

供する。この方法は、まず、前記光学ディスクドライブが、前記ATA/IDEインターフェースから信号を受信し、そして入力/出力デコードを行って、前記ホストコンピュータシステムが前記ファームウェアのアップグレードを要求しているか、又は前記光学ディスクドライブの通常動作を維持することを要求しているかを決定する。その後、前記光学ディスクドライブが、前記ホストコンピュータシステムがファームウェアのアップグレードを要求したときに、前記ATA/IDEインターフェースを前記メモリデバイスに接続し、前記ホストコンピュータが前記メモリデバイスに直接書き込むことによるファームウェアアップグレード動作の実行を許容し、そして前記ホストコンピュータシステムが前記光学ディスクドライブの通常動作を要求したときに、前記マイクロコントローラを前記メモリデバイスに接続する。

【0016】

【発明の実施の形態】本発明の他の目的、特徴及び利点は、好ましいが限定されるものでない実施例の詳細な説明によって明らかにされる。この説明は添付した図面を参照して行われる。

【0017】現在のパーソナルコンピュータに対し、CD-ROMドライブは現在試用されている光学ディスクドライブの大半を占めているが、この明細書で使用する“光学ディスクドライブ”という用語は、同じCDファミリーを起源として近く出現するDVDドライブのような他のタイプの光学ディスクドライブをも包含する。

【0018】一方、ホストコンピュータシステムの通信するために現在のCD-ROMドライブには数種類のインターフェース基準が適用されているが、主たるものはATA/IDE (Intelligent Disk Electronics または Industrial Disk Electronics) 基準か、その強化バージョンEIDEである。他の基準には、SCSI (Small Computer System Interface) とパラレルポートインターフェースが含まれる。本発明による光学ディスクドライブのためのファームウェアコードをアップグレードする装置は、しかしながら、ATA/IDEインターフェースを使用するこれらのドライブへの適用に集中する。

【0019】本発明を説明する目的のために、従来のIDE CD-ROMドライブ用の概略制御回路配置が以下で検証される。図1に示すように、ATA/IDEインターフェースを伴う従来のCD-ROMは、IDEバス10を介してホストコンピュータシステムと通信するための制御回路20を内蔵する。

【0020】一般的なCD-ROMドライブ用の制御回路20は、ドライブの内部動作を制御するマイクロコントローラ(μC)23を備える。この制御回路20は更に、指示コードとドライブファームウェアプログラムのデータとを格納するための不揮発性メモリ(NVM)23を備える。図示したように、典型的なCD-ROMド

ライブはファームウェアコードを図中一方向の矢印で示したように、リードアクセスを介してのみマイクロコントローラ23に供給する。制御回路20は、マイクロコントローラ23とホストコンピュータシステムがインターフェースIDEバス10を介して相互動作することを許容するためのCD-ROMデコーダ(CD-ROM DEC)21を備える。典型的な制御回路20は、例えば、CD-ROMドライブ動作として重要な読出しキャッシュ機能に使用されるダイナミック型ランダムアクセスメモリ(DRAM)25によって提供される作業用メモリ領域を有する。

【0021】当業者にとっては良く知られている様に、制御回路20は更に、レーザピックアップヘッド用のサーボ機構制御部(MECH SERVO)と、アクセスしたCDデータ用のデジタルシグナルプロセッサ(DSP)と、デジタル音楽をアナログ信号に変換するためのデジタル・アナログ変換器(DAC)と、CD-ROMが音楽CDプレーヤとして作用するとき前記アナログ信号を増幅するための前置増幅器(PREAMP)とを備える。これらの機能ブロックの回路は、それらが本発明の主要事項でなく、またそれを改良するものでないので、制御回路20内では単にボックス29として示されている。

【0022】図2は、典型的なIDE CD-ROMドライブの一例として、マイクロコントローラ23、CD-ROMデコーダ21、不揮発性メモリ24相互間の内部接続回路を示している。当業者には良く知られているように、CD-ROMドライブのマイクロコントローラ23は、ドライブユニット内の通常のデータアクセス動作をホストコンピュータシステムの制御下で進行する。これは、CD-ROMドライブソフトウェアを実行し、そしてIDEインターフェースバス10の基準IDE信号のセットを介してドライブマイクロコントローラ23と通信を行うホストプロセッサによって行われる。このバスインターフェース信号のセットには、デバイスアドレス信号HA0-2と、入力/出力ポート選択信号HC SO-1と、16ビットのデータバスHD0-15と、他の制御信号としてリード/ライトストロブ信号HRD/HWRと、割込要求/確認信号HIRQ/HDRAQ/HDACKと、リセット信号RESETと、16ビットI/O転送ステータス指示信号IOCS16とが含まれる。

【0023】本発明に係るATA/IDEインターフェースを介して光学ディスクドライブのファームウェアコードをアップグレードするための装置は、CD-ROMドライブに適用するに好ましい1つの実施例として、図3に示される様な制御回路を構成することを許容する。これは図1に示した様な従来の制御回路をベースとしたものである。

【0024】本発明に係るATA/IDEインターフェー

スを介して光学ディスクドライブのファームウェアコードをアップグレードするための装置の基本概念は、ATA/IDEインターフェースを介してホストコンピュータから出される指示をデコードすることである。デコードされた結果は、ホストコンピュータシステムが、CD-ROMドライブの通常のデータアクセス動作の維持を要求しているのか、あるいはコード格納メモリに書込むことによってオンサイトでファームウェアコードをアップグレードすることを命令しているのかを決定する。ホストコンピュータシステムがオンサイトのファームウェアのアップグレードを命令する場合は、CD-ROMドライブのATA/IDEインターフェースはメモリデバイスに接続され、ホストコンピュータはメモリデバイスへの新たなコードの書込みが実行できるようになる。一方、ホストコンピュータがCD-ROMドライブユニットの通常動作モードへの復帰を要求する場合は、ドライブユニットのマイクロコントローラはメモリデバイスに再接続され、通常のプログラムコードを格納する。

【0025】好ましい実施例では、ATA/IDEインターフェースを介したCD-ROMドライブ用ファームウェアのアップグレードを行うために、マイクロコントローラとそのファームウェアの格納用メモリデバイスとの間に、制御信号マルチプレクサ(多重化回路)とそれに関連する制御ロジック(論理回路)を挿入して使用する。マルチプレクサはドライブのファームウェアメモリデバイスに向かう2つの可能なデータアクセス経路の1つを提供するために使用される。経路の1つは光学ディスクドライブの通常動作条件下で設定され、そこではマルチプレクサはメモリデバイスをマイクロコントローラに接続し、マイクロコントローラが実行するためにメモリからファームウェアプログラムのコードとデータを読み取り可能になる。明らかなように、メモリデバイスに対しマイクロコンピュータによって実行されるメモリアクセス動作のこのモードは一方方向性であり、マイクロコントローラはメモリデバイスに対し読出しアクセスだけを実行する。

【0026】ファームウェアコード格納メモリに対し設定された他方の経路は、メモリデバイスそれ自体が、コードアップグレードが命令されたときにホストコンピュータシステムから直接アクセスされるようにするためのものである。この動作モードでは、マルチプレクサはメモリデバイスを光学ディスクドライブユニットのIDEインターフェースに接続し、ホストシステムのプロセッサがファームウェアメモリのメモリ空間に直接アクセスできるようになる。この直接アクセスには、メモリデバイスへの書込みと、メモリデバイスからの読出しとがある。メモリデバイスからの読出し能力は、正しいプログラムの内容が書込まれたか否かを確認する手段として必要である。

【0027】ここで述べられるオンサイトのアップグレ

ード動作とは、光学ディスクドライブの内部からメモリデバイスを取り外す必要なく、ファームウェアコード格納メモリデバイスに対しアップグレード動作を行うことと定義される。このアップグレード動作には煩雑なプログラミングツールは必要とされない。

【0028】かくして、図3に示されるように、ATA/IDEインタフェースを介して光学ディスクドライブのファームウェアコードをアップグレードするための本発明の装置は、ドライブ制御電気回路30内のマイクロコントローラ33とファームウェアコードメモリ34との間に配置されたマルチプレクサ(MUX)36を備える。CD-ROMデコーダ31によって行われるデコード処理に加えて、メモリデバイス34を通常の動作モード用にマイクロコントローラ33に接続するか、あるいはファームウェアのアップグレード動作モード用にホストプロセッサに接続するためのIDEバス10に接続するかを、マルチプレクサ36に選択させるためのプログラミングコントローラ(PRG CNTL)32が使用される。このことを達成するために、図3に示されているマルチプレクサ36は、その詳細を後述するように、基本的に2対1の多重化デバイスである。

【0029】それ故、プログラミングコントローラ32がドライブのファームウェアデバイス34とマイクロコントローラ33に接続すると、両者は周知の手法で動作するようになり、図1の回路配置の場合のように、マイクロコントローラ33はメモリ34からファームウェアコードをアクセスする通常動作を実行する。換言すれば、プログラミングコントローラ32がメモリデバイスをドライブマイクロコントローラ33に接続するので、マイクロコントローラ33はその実行コードをメモリデバイス33から回収する読出し専用アクセスを実行する。

【0030】一方、プログラミングコントローラ32がメモリデバイス34を光学ディスクドライブユニットのIDEバス10に接続すると、ホストコンピュータシステムのプロセッサはメモリデバイス34へのアクセスを実行することが許容される。メモリデバイス34に対する読出しおよび書込みの両アクセスは、メモリデバイスにおける読出し/書込み動作のシリーズにおけるオンサイトのファームウェアコードのアップグレードの促進を許容される。これは、IBM互換コンピュータシステムに対するシステムBIOS(Basic Input/Output System)コードをオンボード(on-board)でアップグレードすることに類似した動作である。オンボードのアップグレード動作では、BIOSコードが存在するメモリデバイスは、コンピュータシステムのマザーボードから取外される必要がなく、また煩雑な半導体メモリプログラム用装置を必要としない。いくつかのスイッチの適切な設定により、ソフトウェアプログラムを実行するコンピュータシステムによって、コードアップグレード動作

は完全に実行され得る。

【0031】図4を参照する。図4は、ファームウェアコードのオンサイトのアップグレード動作を実行するための本発明の装置を使用したCD-ROMドライブデバイス用の制御回路の一実施例を示す模式図である。図4の回路において、通常動作を実行するために、あるいはアップグレード動作を行うために、メモリデバイス34をマイクロコントローラ33またはドライブのIDEバスに選択的に接続する制御が実行される。

【0032】コントローラを透視することから、図4において囲まれたプログラミングコントローラ32の回路は、ファームウェアコードのアップグレードが命令されたとき、IDEインターフェースを介してホストコンピュータシステムから出された指示とデータを、メモリデバイス34で必要な電気信号に変換し、そのメモリ内容をプログラミングすることを許容する。一方、このプログラミングコントローラ32の回路はまた、通常のドライブ動作が要求されたときに、マイクロコントローラ33とメモリデバイス34との通常の接続を許容する。

【0033】指示とデータのコードアップグレードを達成するために、図4で囲まれたプログラミングコントローラ32は、I/Oアドレスデコーダ(I/O ADDRDEC)321と、アップグレード初期化キーイネーブル論理回路(INITKEY EN)322と、アドレスプログラムイネーブルラッチ(APEN)323と、データプログラムイネーブルラッチ(DPEN)324と、制御プログラムイネーブルラッチ(CPEN)325とを備える。

【0034】オンサイトのアップグレード動作が要求されると、ホストシステムは一連の対応する指示及びデータをIDEバス10に出し、プログラミングコントローラ32がバス10自体から独立して動作できるようにする必要がある。これらの指示及びデータは、ドライブが通常の動作モードにあるときは決してIDEバス10上に存在してはならない。その代わりに、これらはオンサイトのアップグレードが要求されたときだけ出現して、プログラミングコントローラ32の初期化に使用される。

【0035】例えば、IBM互換システムの場合に使用される好ましい実施例では、ホストシステムのプロセッサは、I/Oアドレス1F5において、一連の連続し、且つ予め定義された32バイトの情報を書き込むことができる。図4に示すように、これらのデータバイトは、I/Oアドレスデコーダ321によってデコードされる時、I/Oアドレス1F5を介してバス10A上で受信される。I/Oアドレス1F5を介して受信されたデータは、それからアップグレード初期化キーイネーブル論理回路322に送信される。バス10Aは、制御回路30内部のIDEバス10の延長部である。

【0036】アップグレード初期化キーイネーブル論理

回路322は、I/OアドレスIF5の論理的に正の信号によって起動され、連続した32バイトの情報を受信する。これらの情報バイトが所定のデータのセットと比較され、互換性があると判断されたときは、アップグレード初期化キーイネーブル論理回路322はアップグレード初期化キーイネーブル信号KEYOKを生成する。この信号KEYOKは、アドレスプログラムイネーブルラッチ323と、データプログラムイネーブルラッチ324と、制御プログラムイネーブルラッチ325とに、それぞれ中継される。その後、これら3つのラッチは、制御回路30内で同時にイネーブル状態になる。

【0037】一方、アドレスプログラムイネーブルラッチ323は、バス10A上の16ビットのIDEデータ信号HD0-15をラッチし、バス3236に出力する。バス3236上の対応する16ビットのデータ信号は、信号IA0-15と呼ばれる。一方、データプログラムイネーブルラッチ324も、バス10A上の同じ16ビットのIDEデータ信号HD0-15の下位バイト(8ビット)をラッチし、バス3236上の8ビットのデータ信号IHD0-7として出力する。同様の手法で、制御プログラムイネーブルラッチ325は、バス10A上の2バイトのIDEデータ信号HD0-15の下位8ビットをラッチし、バス3236に出力する。ラッチされたビット中から選択されたビットは、ライトイネーブル信号IWE、チップイネーブル信号ICE、出力イネーブル信号IOE、IDEプログラミングイネーブル信号IDEPENとして、プログラミング動作を実行するためにメモリデバイス34をストロブする必要があるときに使用される。

【0038】例えば、IDEプログラミングイネーブル信号IDEPENが論理的に正のステータスになると、ホストシステムのプロセッサはメモリデバイス34をプログラムするのに必要な指示とデータを出すことができる。これらの指示とデータは、指定されたI/Oアドレスにおいて、IDEバス10を介してメモリデバイス34に書き込まれる。例えば、図4の実施例では、メモリデバイス34に対するリード/ライトアクセスのアドレスは、IDEインターフェース上のI/OアドレスIF0においてデータ信号HD0-15中に出される。一方、I/OアドレスIF3とデータビットHD0-7は、メモリデバイス34に書き込まれるプログラムコードを指定されたアドレス空間に中継するために使用される。更に、I/OアドレスIF4とデータビットHD0-7は、プログラムされるメモリデバイス34が要求したときに、必要なデバイスストロブ信号OE、CD、WEを中継するために使用できる。CD-ROMドライブ全体の状態が通常の動作モードであるかアップグレードプログラミング動作モードであるかを示すために使用されるプログラミングイネーブル信号IDEPENは、それ自体がデータHD0-7中で指定されたビットであ

るように、I/OアドレスIF4を介して中継される。

【0039】当業者には知られているように、16ビットのアドレスはCD-ROMドライブに通常搭載されているファームウェアメモリデバイスをアドレス指定するには十分である。例えば、8ビットメモリの場合、IA0-15信号線を介して中継される16ビットのアドレスは、64キロバイトのメモリ空間をアドレス指定する事ができる。コスト低減を考慮して、通常8ビットのメモリデバイスがファームウェア格納デバイスとして使用される。このことは、最近の高性能パーソナルコンピュータシステムのホストコンピュータに比べて、CD-ROMドライブのような光学ディスクドライブが相対的に低速の周辺機器であることから、都合がよい。マイクロコントローラ33(図3)に接続されたIHD0-7信号線を介して中継される8ビットのデータ経路の使用は、一般に通常のCD-ROMドライブに対する配置としては十分である。更に、3つのメモリデバイス制御イネーブル信号IOE、IWE、ICEの全ては、プログラミングイネーブル信号IDEPENと同様に、IDEインターフェースバス上を中継されるHD0-7データビットを介して全て得ることができる。

【0040】再び図3を参照すると、プログラミングコントローラ32の回路が、16個のアドレス信号IA0-15、8個のデータ信号IHD0-7、4個の制御信号ICE、IOE、IWE、IDEPENを、出力端としてのバス3236上に転送することが示されている。これらの信号がバス3236上に転送されると、これらは実効的にマルチプレクサ36の入力となり、そこではメモリデバイス34に対するアクセス信号のセットを選択的に接続するために、プログラミングイネーブル信号IDEPENの論理的に正のステータスに依存して多重化動作を促進する。適切なタイミング制御の下で、この中継された信号のセットは、CD-ROMドライブそれ自体のマイクロコントローラ33でCD-ROMデータ読み出し期間にドライブを通常動作させるために使用されるか、あるいはホストコンピュータシステムのプロセッサでファームウェアアップグレード期間に使用される。勿論、この場合にファームウェア格納部として使用される半導体メモリデバイス34は、電氣的に再プログラミングできるものでなければならない。それは、例えばEEPROMである。

【0041】以上の説明は、本発明により構築されるCD-ROMドライブの電子回路の動作中の、メモリデバイス中のファームウェアコードのアップグレードに集中している。一方、通常の状態では、ファームウェアコードのアップグレードは、CD-ROMドライブの寿命期間に数回もないと予測される希な動作であることが明記される。多くの場合、CD-ROMドライブは、ドライブに挿入されたCD-ROMの記憶内容を読み取る。この場合、図4のプログラミングコントローラ32内のア

プログラミングイネーブル信号IDEPENは、メモリデバイス34を実質的に直接マイクロコントローラ33に接続し、マイクロコントローラ33がCD-ROMドライブの機能を遂行するメモリデバイス34内のコードを実行できるようにする。この意味で、本発明の装置を使用するCD-ROMドライブは、実質的に従来のドライブと同様になる。

【0042】かくして、図3のブロック図に示すように、ホストコンピュータシステムは、標準IDEバス10を介してメモリデバイスに対し直接、そのファームウェアアップグレード用のコード書き込み動作を遂行することができる。プログラミング動作は、指示と関連するデータとをバス10Aを介して、それからプログラミングコントローラ32の制御下のバス3236を介して出すホストコンピュータによって遂行される。更新されたデータはその後、マルチプレクサ36によって切替えられたときに、バス3436を介してメモリデバイス34に送信される。図5は、マルチプレクサ36の好ましい実施例の回路配置と動作の詳細を示している。

【0043】図5に示すように、図3のブロック図で使用されたマルチプレクサ36は、5つの2対1マルチプレクサユニット361、362、363、364、365のグループを有する。5つのマルチプレクサユニットの全ては、同じ制御信号で切替えられる。図3及び図4で囲まれた実施例の場合、プログラミングイネーブル信号IDEPENは制御信号として使用される。5つのマルチプレクサユニットは全てそれぞれバス3336に、それからCD-ROMドライブのマイクロコントローラ33に接続された入力第1のセットを有する。5つのマルチプレクサユニットの入力第2のセットは、バス3336に、それからプログラミングコントローラ32のを経由した接続を介してIDEバス10に接続される。5つのマルチプレクサユニットの多重化出力は、バス3436に、それからメモリデバイス34に接続される。図3に明瞭に囲まれているように、このバス接続配置は、プログラミングイネーブル信号IDEPENが、通常動作時にはマイクロコントローラ33側に、そしてファームウェアアップグレード動作時にはIDEバス10側に接続されるメモリデバイス34の制御を遂行することを許容する。

【0044】マルチプレクサユニット361は、メモリデバイス34のプログラミング期間にアドレスビットを提供する16ビットのIA0-15信号を中継する必要性に便宜を図るために、16ビットの切換え可能な信号経路幅を有する。CD-ROMドライブが通常動作しているとき、この16ビットの信号経路幅は、中継される16ビットのマイクロコントローラアドレスビット $\mu$ CA0-15を、メモリデバイス34の16ビットのアドレスFA0-15に接続する事を許容する。同様に、マルチプレクサユニット362は8ビット幅の切換え可能

な信号経路を有し、2つのセットのデータ線IHD0-7と $\mu$ CD0-7を、メモリデバイス34の対応する8ビットのデータバスFD0-7に中継する必要性に便宜を図るようにする。これとは対照的に、マルチプレクサ363、364、365は、メモリデバイス34の制御信号WE、CE、OEの中継に使用される単線の多重化回路である。

【0045】かくして、ATA/IDEインターフェースを伴い、オンサイトのファームウェアコードアップグレード用の装置を使用するCD-ROMは、2つの動作モードの中の必要性が生じた1つで動作する。通常モードの動作では、CD-ROMドライブのファームウェアコードメモリデバイスとマイクロコントローラは、通常のCD-ROM動作に誘導するようにマイクロコントローラがメモリ内のコードをアクセスできるようにするために、互いに接続することができる。一方、CD-ROMドライブがファームウェアアップグレードモードにあるときは、メモリデバイスはドライブのIDEインターフェースに接続され、ホストコンピュータシステムのプロセッサが、IDEバスに接続されたコンピュータシステムバスを介して、メモリデバイス内のアドレス指定したメモリ空間に直接書き込みできるようにする。プログラミング指示コードと新たなコード内容はCD-ROMドライブデバイス内に装備されたIDEインターフェースを介してメモリデバイスに中継できる。

【0046】この直接オンサイトのプログラミング能力と共に、本発明の回路装置を使用するCD-ROMドライブは、ファームウェアコードのオンサイトのアップグレード動作の大きな利便性を享受することができる。このドライブ自体はコンピュータシステムのキャビネット内部から取り出す必要がない。コードアップグレード動作全体は、完全にソフトウェア制御で誘導できる。適切な情報が与えられることにより、一般的なエンドユーザでも、例えばコンピュータシステムの表示画面上に示される詳細な指示に基づいて、アップグレード動作を行うことができる。CD-ROMドライブの製造者にとっては、全ての要求するユーザに対し通常の郵便または電子メールサービスあるいはインターネットのような公衆通信網のファイル転送サービスを利用して、ユーザアップグレード用のソフトウェアとファームウェアプログラムコードの新バージョンの双方を入手させることができる利点がある。製造者とユーザの双方にとって、資源と時間の効果的なコスト保全が期待できる。

【0047】更に、CD-ROMドライブ用のATA/IDEインターフェースを介してオンサイトのファームウェアコードのアップグレードを実行する装置であるから、この装置はドライブ回路内のASICチップセットに集積化され得る。事実、当業者から評価され得るように、本発明の装置は、例えば図3のCD-ROMデコーダ31内に、あるいはマイクロコントローラ自体の中に

さえも集積化され得る。

【0048】本発明が例を示して、そして好ましい実施例に関して説明されたが、本発明は開示した実施例に限定されるものでないことが理解されるべきである。むしろ、本発明は、添付した請求の範囲の精神に含まれる種々の変形と同様の配置を包含することを意図し、この請求の範囲は全てのその様な変形と同様の構成が包含されるように、最も広い解釈が与えられるべきものである。

【図面の簡単な説明】

【図1】 従来のCD-ROMドライブの電気的コントローラの回路を示すブロック図である。

【図2】 図1のCD-ROMドライブにおける典型的なドライブIDEインターフェースに関連した、マイクロコントローラ、ファームウェアメモリ及びCD-ROMドライブデコーダ間の従来のインターフェース接続を示す図である。

【図3】 本発明の好ましい実施例に関連したファームウェアコードをオンサイトでアップグレードするための装置を使用したCD-ROMドライブの電気的なコントローラの回路を示すブロック図である。

【図4】 図3の電気的回路配置を使用したCD-ROM

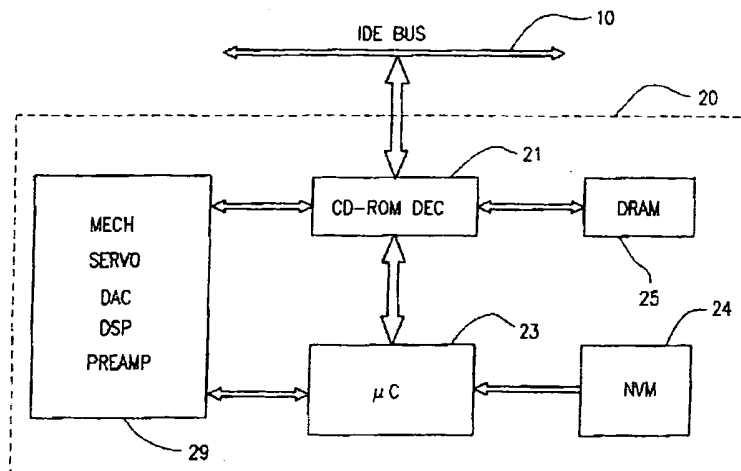
Mドライブのアプログラミングコントローラを模式的に示す図である。

【図5】 図3の電気的回路配置を使用したCD-ROMドライブのマルチプレクサを模式的に示す図である。

【符号の説明】

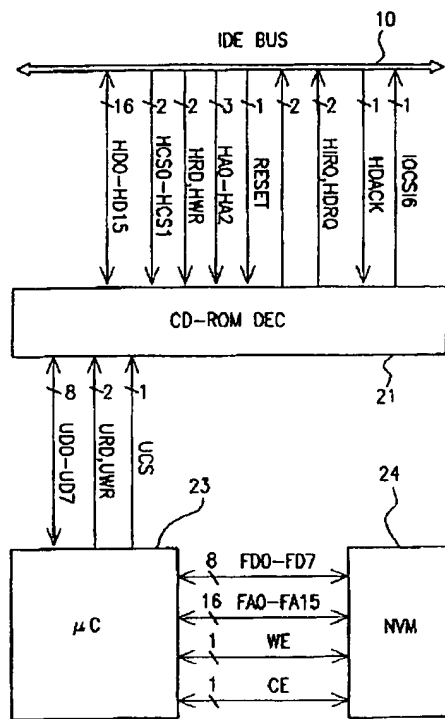
- 10 IDEバス
- 30 ドライブ制御電気回路
- 31 CD-ROMデコーダ
- 32 アプログラミングコントローラ
- 321 I/Oアドレスデコーダ
- 322 アップグレード初期化キーネーブル論理回路
- 323 アドレスプログラムイネーブルラッチ
- 324 データプログラムイネーブルラッチ
- 325 制御プログラムイネーブルラッチ
- 33 マイクロコントローラ
- 34 ファームウェアコードメモリ（不揮発性メモリ）
- 35 DRAM
- 36 マルチプレクサ
- 361～365 マルチプレクサユニット

【図1】



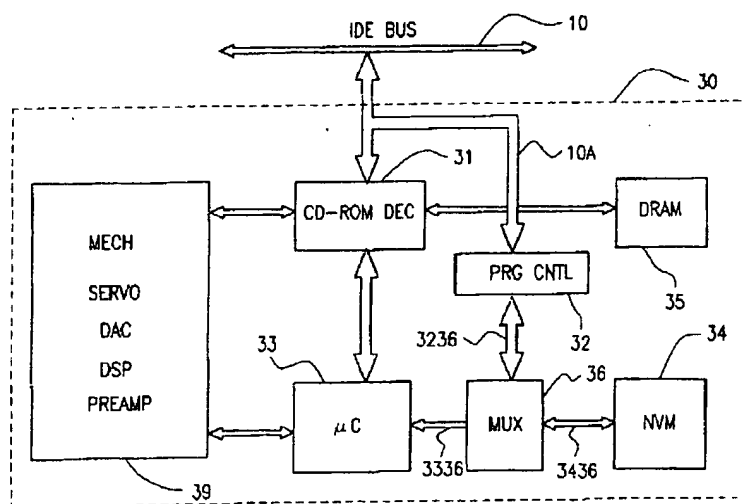
先行技術

【図2】

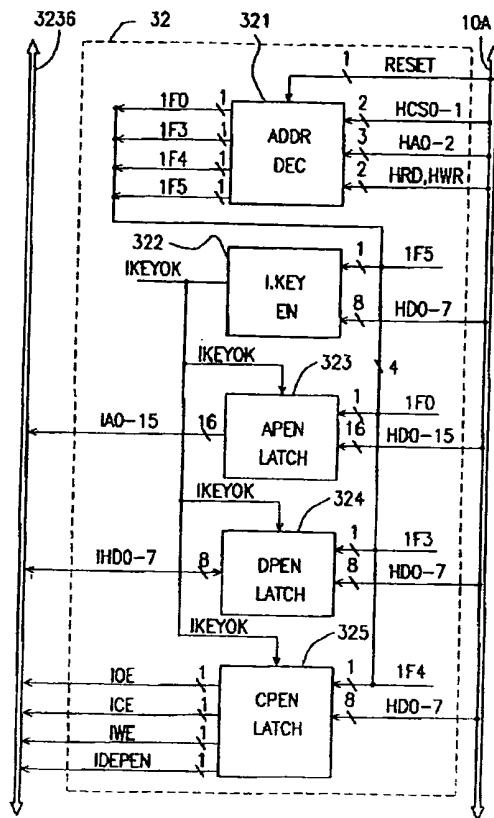


先行技術

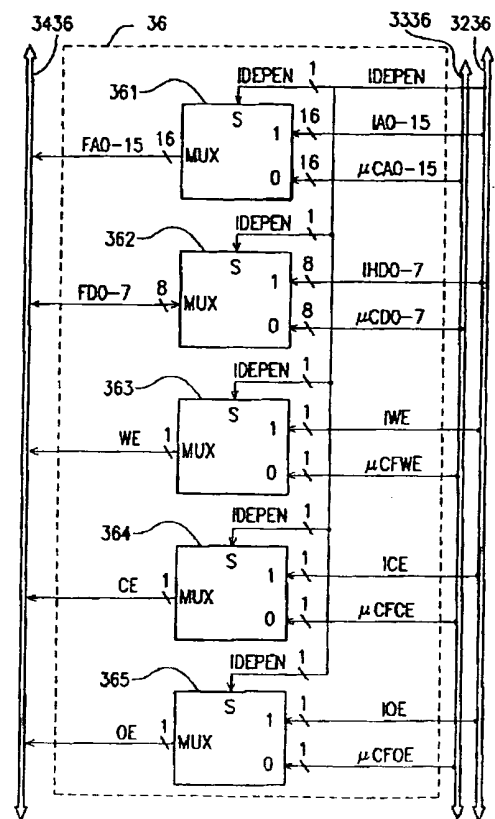
【図3】



【図4】



【図5】





## 【 外 國 語 明 細 書 】

## 1 . T i t l e   o f   I n v e n t i o n

METHOD AND APPARATUS FOR UPGRADING FIRMWARE CODE FOR  
OPTICAL DISK DRIVE VIA ATA/IDE INTERFACE

## 2 . C l a i m s

1. An apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive comprises a memory device for storing the firmware code, a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive, and a drive decoder for decoding to connect the optical disk drive to a host computer system via the ATA/IDE interface, the apparatus comprising:

a programming controller means, receiving signals from the ATA/IDE interface and performing input/output decode for generating a programming enable signal that determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation; and

a multiplexer means, having a first and a second input, a multiplexed output and a multiplexing select input, the first input being connected to the programming controller means, the second input being connected to the microcontroller, and the multiplexing select input receiving the programming enable signal, the multiplexer means switching the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests for a firmware upgrade.

2. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 1, wherein the programming controller means further comprising:

an input/output decoder, receiving signals from the ATA/IDE interface and performing the input/output decode for generating the programming enable signal and further generating a first, a second, a third and a fourth enable signal;

an initialization key enable logic, receiving the first enable signal and the corresponding signals from the ATA/IDE interface for generating an upgrade initialization key signal when the host computer system requests for a firmware

upgrade;

an address program enable latch, receiving the upgrade initialization key signal, the second enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written address signals for the memory device when the host computer system requests for a firmware upgrade;

a data program enable latch, receiving the upgrade initialization key signal, the third enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written data signals for the memory device when the host computer system requests for a firmware upgrade; and

a control program enable latch, receiving the upgrade initialization key signal, the fourth enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the writing control signals for the memory device when the host computer system requests for a firmware upgrade.

3. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 2, wherein the corresponding signals received by the address program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises the total 16-bit address signals of the ATA/IDE interface.

4. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 2, wherein the corresponding signals received by the data program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

5. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 2, wherein the corresponding signals received by the control program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

6. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 2, wherein the optical disk drive is a CD-ROM drive.

7. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 2, wherein the optical disk drive is a DVD drive.

8. An apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive comprises a memory device for storing the firmware code, a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive, and a drive decoder for decoding to connect the optical disk drive to a host computer system via the ATA/IDE interface, the apparatus comprising:

a programming controller means, receiving signals from the ATA/IDE interface and performing input/output decode for generating a programming enable signal that determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation, the programming controller means comprises:

an input/output decoder, receiving signals from the ATA/IDE interface and performing the input/output decode for generating the programming enable signal and further generating a

first, a second, a third and a fourth enable signal;

an initialization key enable logic, receiving the first enable signal and the corresponding signals from the ATA/IDE interface for generating an upgrade initialization key signal when the host computer system requests for a firmware upgrade;

an address program enable latch, receiving the upgrade initialization key signal, the second enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written address signals for the memory device when the host computer system requests for a firmware upgrade;

a data program enable latch, receiving the upgrade initialization key signal, the third enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written data signals for the memory device when the host computer system requests for a firmware upgrade; and

a control program enable latch, receiving the upgrade initialization key signal, the fourth enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the writing control signals for the memory device when the host computer system requests for a firmware upgrade; and

a multiplexer means, having a first and a second input, a multiplexed output and a multiplexing select input, the first input being connected to the programming controller means, the second input being connected to the microcontroller, and the multiplexing select input receiving the programming enable signal, the multiplexer means switching the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the

host computer system requests for a firmware upgrade.

9. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 8, wherein the corresponding signals received by the address program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises the total 16-bit address signals of the ATA/IDE interface.

10. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 8, wherein the corresponding signals received by the data program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

11. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 8, wherein the corresponding signals received by the control program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

12. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 8, wherein the optical disk drive is a CD-ROM drive.

13. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 8, wherein the optical disk drive is a DVD drive.

14. A method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive comprises a memory device for storing the firmware code and a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive; the method comprising the steps of:

the optical disk drive receiving signals from the ATA/IDE interface and performing input/output decode determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation; and

the optical disk drive connecting the ATA/IDE interface to the memory device, allowing for a host computer to perform a firmware upgrade operation by directly writing into the memory device when the host computer system requests for a firmware upgrade, and connecting the microcontroller to the memory device when the host computer system requests for the normal operation of the optical disk drive.

15. The method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 14, wherein the optical disk drive is a CD-ROM drive.

16. The method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 14, wherein the optical disk drive is a DVD drive.

17. A method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive comprises a memory device for storing the firmware code, a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive, a drive decoder for decoding to connect the optical disk drive to a host computer system via the

ATA/IDE interface, a multiplexer means and a programming controller means, the multiplexer means having a first and a second input, a multiplexed output and a multiplexing select input, the first input being connected to the programming controller means and the second input being connected to the microcontroller, and the programming controller means selectively switching the multiplexing of the multiplexer means; the method comprising the steps of:

the programming controller means receiving signals from the ATA/IDE interface and performing input/output decode for generating a programming enable signal that determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation; and

the multiplexer means having the multiplexing select input receiving the programming enable signal for switching the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests for a firmware upgrade.

18. The method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 17, wherein the step of the programming controller means generating the programming enable signal further comprising the steps of:

an input/output decoder of the programming controller means receiving signals from the ATA/IDE interface and performing the input/output decode for generating the programming enable signal and further generating a first, a second, a third and a fourth enable signal;

an initialization key enable logic of the programming controller means receiving the first enable signal and the corresponding signals from the ATA/IDE interface for generating an upgrade initialization key signal when the host computer system requests for a firmware upgrade;

an address program enable latch of the programming controller means receiving the upgrade initialization key signal, the second enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written address signals for the memory device when the host computer system requests for a firmware upgrade;

a data program enable latch of the programming controller means receiving the upgrade initialization key signal, the third enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written data signals for the memory device when the host computer system requests for a firmware upgrade; and

a control program enable latch of the programming controller means receiving the upgrade initialization key signal, the fourth enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the writing control signals for the memory device when the host computer system requests for a firmware upgrade.

19. The method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 18, wherein the corresponding signals received by the address program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises the total 16-bit address signals of the ATA/IDE interface.

20. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 18, wherein the corresponding signals received by the data program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.



21. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 18, wherein the corresponding signals received by the control program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

22. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 18, wherein the optical disk drive is a CD-ROM drive.

23. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 18, wherein the optical disk drive is a DVD drive.

24. A method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive comprises a memory device for storing the firmware code, a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive, a drive decoder for decoding to connect the optical disk drive to a host computer system via the ATA/IDE interface, a multiplexer means and a programming controller means, the multiplexer means having a first and a second input, a multiplexed output and a multiplexing select input, the first input being connected to the programming controller means and the second input being connected to the microcontroller, and the programming controller means selectively switching the multiplexing of the multiplexer means; the method comprising the steps of:

the programming controller means receiving signals from the ATA/IDE interface and performing input/output decode for generating a programming enable signal that determines whether the host computer system has

requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation, the generation of the programming enable signal further comprising the steps of:

an input/output decoder of the programming controller means receiving signals from the ATA/IDE interface and performing the input/output decode for generating the programming enable signal and further generating a first, a second, a third and a fourth enable signal;

an initialization key enable logic of the programming controller means receiving the first enable signal and the corresponding signals from the ATA/IDE interface for generating an upgrade initialization key signal when the host computer system requests for a firmware upgrade;

an address program enable latch of the programming controller means receiving the upgrade initialization key signal, the second enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written address signals for the memory device when the host computer system requests for a firmware upgrade;

a data program enable latch of the programming controller means receiving the upgrade initialization key signal, the third enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE interface to generate the written data signals for the memory device when the host computer system requests for a firmware upgrade; and

a control program enable latch of the programming controller means receiving the upgrade initialization key signal, the fourth enable signal, and the corresponding signals from the ATA/IDE interface for latching the corresponding signals on the ATA/IDE

interface to generate the writing control signals for the memory device when the host computer system requests for a firmware upgrade; and the multiplexer means having the multiplexing select input receiving the programming enable signal for switching the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests for a firmware upgrade.

25. The method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 24, wherein the corresponding signals received by the address program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises the total 16-bit address signals of the ATA/IDE interface.

26. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 24, wherein the corresponding signals received by the data program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

27. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 24, wherein the corresponding signals received by the control program enable latch from the ATA/IDE interface when the host computer system requests for a firmware upgrade comprises at least the least significant eight bits of the data signals of the ATA/IDE interface.

28. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 24, wherein the optical disk drive is a CD-ROM drive.

29. The apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface of claim 24, wherein the optical disk drive is a DVD drive.

### 3 . D e t a i l e d   E x p l a n a t i o n   o f   t h e   I n v e n t i o n

## BACKGROUND OF THE INVENTION

### Field of the Invention

The invention relates in general to on-site upgrading of firmware code in optical disk drives. In particular, the invention relates to an apparatus for optical disk drives and its corresponding method capable of implementing firmware code upgrade via the ATA/IDE interface of the host computer system.

### Technical Background

Optical disk drives, in particular the category of CD-ROM drives, have become the digital optoelectronic memory storage equipment so popular that almost every personal computer system has at least one installed. In the world of personal computers, the design of CD-ROM drives was intended originally for the storage of mass information on the CD-ROM discs. The concept, in a sense, is basically the same as the magnetic disk drives, which store information on the surface of the storage media. No writing capability is available to the user for this member of the CD family, and thus the name "CD-ROM." Large data is, however, accessible from CD-ROM discs that are mass produced in low cost.

Although CD-ROM has been adopted by the ISO as a standard for digital data storage, there are also other applications beyond the simple storage of large amount of data. Under proper software control, a CD-ROM drive may also be used to access several other types of discs in the CD family.

For example, in personal computer systems, CD-ROM drives are frequently used as the traditional CD player for playing back music pieces stored on the very first generation of CD, namely, the music CD. After the introduction of the original music CD, there had emerged in the family such formats as the Video CD (VCD). Conventional CD-ROM drives for personal computer systems are designed to be able to play back VCD video programs on the computer display

screen. Due to the inherent limitation of the MPEG-1 adopted, the VCD standard is not for serious applications, and a couple of groups of electronic appliance conglomerates had recently achieved on a new standard known as the Digital Video Disc (or Digital Versatile Disc).

This is a new video data storage format, a new member of the CD family based on the MPEG-2 standard that provides substantially better video effects via the use of higher resolution. As the DVD standard had proposed, backward compatibility for the new generation of DVD drives to the CD-ROM format has been designed into the new system. In other words, the up-coming DVD drives will be able to read CD-ROM in addition to its default DVD.

Thus, as an optical Compact Disc subsystem for a computer system, the CD-ROM drive can be programmed to access different formats of the CD family of media. In the popular x86-based IBM-compatible machines, this is implemented in a manner known as software driver installation. With the installation of the proper software drivers, the computer system, governed by the operating system, can be used to read data files stored on the CD-ROM, playback music CD, or video VCD. These software drivers are essentially established on top of the hardware-level control program resident in the control system of a CD-ROM drive. This firmware program is usually stored in non-volatile semiconductor memory devices, such as the erasable-programmable read-only memory (EPROM) or electrically erasable-programmable read-only memory (EEPROM).

Based on this firmware, CD-ROM drives can be controlled by the host computer system to suitably implement the functionality of music CD player, VCD player, and/or CD-ROM data drive. However, under certain circumstances, in particular when new models of CD-ROM drives are released to market for sale, software programs, either the firmware in the control electronics of the drive itself or the operating system/application-level driver, may have program problems (program bugs) that might cause the drive to malfunction. If the problem was with the operating system/application-level driver, it's fix may not involve the CD-

ROM drive at all. If, however, the problem was unfortunately with the firmware of the CD-ROM drive, the software fix will inevitably cause many trouble.

Basically, if the drive firmware has any program problem, the firmware stored in the semiconductor memory device must be upgraded, namely, replaced by another copy without problem. If the semiconductor memory used for holding the firmware program was device like EPROM, it must be removed from its inserted socket or de-soldered from the printed circuit board so as to implement reprogramming of its memory content. Normally, this involves ultraviolet irradiation on the EPROM device for its memory content erasure. If the erased memory device was to be reused again, as is usually the case, it then has to be reprogrammed to hold the new problem-free firmware program. Even if an EEPROM device was used to store the drive firmware, it must be handled in the manner as in the case of EPROM. This is because the conventional CD-ROM drive control logic lacks the design for on-site reprogramming.

The above-described procedure of firmware upgrade/fix for the conventional CD-ROM drives requires that the casing of the drive be opened in order to be able to have access to the firmware memory device. Before the memory device can even be accessed, the CD-ROM drive itself must be removed from the drive bay of the computer system unit, if it has already been installed. Further, erasure and programming of EPROM and/or EEPROM devices normally require the use of dedicated equipment such as ultraviolet eraser and programmer. Therefore, typical firmware upgrade operation for a CD-ROM drive is not normally handled by the end user of the computer system. A more likely scenario would be to have service personnel remove the CD-ROM drive from the system, and have the drive sent back to the manufacturer's site where the firmware gets upgraded.

When the CD-ROM drive whose firmware is to be upgraded is returned to the manufacturer's facility, the casing is opened, and the memory device is removed from the drive electronics utilizing proper tools and/or

equipment. After reprogramming, the memory device, or a substitute thereof containing the correct firmware program code, may then be replaced back into the drive electronics. After adequate testing procedures, the upgraded CD-ROM drive is ready to be returned to the owner. Again, qualified personnel must replace the drive back into the computer system. As is obvious, relatively complicated professional knowledge is required to perform all the tasks involved in the upgrade of a CD-ROM drive. In a time when severe competition has driven the prices of CD-ROM drives down to the level where it is difficult for manufacturers to maintain a descent profit margin, such upgrading operation is strictly disastrous.

This is because the necessity to perform firmware upgrade for a CD-ROM drive often arises within the warrantee period of the product. As such, it is convention that the manufacturer will have to be responsible for the expenditures of drive removal from the host computer system as well as shipment, not to mention all the additional man power required to perform the upgrade. On the other hand, in the process of upgrade, the drive and its components may easily be subject to damages to cause additional loss. For the drive owner, the cost to find some technician to remove the drive from the computer system (if the manufacturer was not going to be responsible) may frequently amount to the level of the cost of some low-end CD-ROM drives.

#### SUMMARY OF THE INVENTION

It is therefore and object of the invention to provide an apparatus and its corresponding method for upgrading firmware code on-site for optical disk drives via the ATA/IDE interface of the host computer system without having to remove the drive unit from the computer and opening up the cabinet.

It is another object of the invention to provide an apparatus and its corresponding method for upgrading firmware code on-site for optical disk drives via the ATA/IDE interface of the host computer system by the host processor directly writing into the memory device holding the firmware code.

In order to achieve the above objects, the invention provides an apparatus for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive has a memory device for storing the firmware code, a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive, and a drive decoder for decoding to connect the optical disk drive to a host computer system via the ATA/IDE interface. The apparatus includes a programming controller that receives signals from the ATA/IDE interface and performs input/output decode for generating a programming enable signal that determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation. A multiplexer has a first and a second input, a multiplexed output and a multiplexing select input, the first input is connected to the programming controller, the second input connected to the microcontroller, and the multiplexing select input receives the programming enable signal. The multiplexer switches the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests for a firmware upgrade.

To achieve the above objects, the invention further provides a method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface, the optical disk drive has a memory device for storing the firmware code and a microcontroller executing the firmware code for controlling the optical disk content access operation of the drive. The method includes the steps of first having the optical disk drive receive signals from the ATA/IDE interface and perform input/output decode to determine whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation. Then the optical disk drive connects the ATA/IDE interface to the memory device, allowing for a host computer to perform a firmware upgrade operation by directly writing into the memory device when the host computer system requests for a firmware upgrade, and connects the microcontroller to the



memory device when the host computer system requests for the normal operation of the optical disk drive.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Note that although for present-day personal computers, CD-ROM drives constitute the majority of optical disk drives currently in service, however, the term "optical disk drive" referred to in this specification should include other types of optical disk drives such as the emerging DVD drive which also originates from the same CD family.

On the other hand, although there are several interface standards currently adopted by conventional CD-ROM drives for communicating with their

host personal computer systems, however, the majority being the ATA/IDE (Intelligent Disk Electronics or Industrial Disk Electronics) standard, or its enhanced version EIDE. Other standards include the SCSI (Small Computer System Interface) and even the parallel port interface. The apparatus for upgrading firmware code for optical disk drive of the invention, however, concentrates on the application to those drives with the ATA/IDE interface.

For the purpose of describing the invention, the general control circuitry configuration for a conventional IDE CD-ROM drive is examined below. As is illustrated in FIG. 1, the conventional CD-ROM drive with ATA/IDE interface has the built-in control electronics generally identified by the reference numeral 20 communicating with the host computer system via the IDE bus 10.

Control circuitry 20 for the typical CD-ROM drives includes a microcontroller ( $\mu$ C) 23, which is responsible for coordinating the internal operation of the drive. The circuitry 20 further includes a non-volatile memory device (NVM) 24 for the storage of the instruction code and data of the drive firmware program. As is seen in the drawing, the typical CD-ROM drive provides the firmware code to the microcontroller 23 only via read access, as is signified by the single-directional arrow in the drawing. The circuitry 20 has a CD-ROM decoder (CD-ROM DEC) 21, which allows for the interacting operation between the microcontroller 23 and the host computer system via the interface IDE bus 10. A typical circuitry 20 further has a working memory space, which can be provided by, for example, a dynamic random-access memory (DRAM) 25 that serves one important function of read caching as the CD-ROM drive operates.

As persons skilled in the art are well familiar, the control circuitry 20 is further complete with the mechanism servo control (MECH SERVO) for the laser pick-up head, the digital signal processor (DSP) for the accessed CD data, as well as the digital-to-analog converter (DAC) for converting the digital music data into analog signal and the preamplifier (PREAMP) for the amplification of this analog music signal when the CD-ROM is operated as a music CD player. Circuitry of

these functional blocks are simply represented by a box 29 in the circuitry 20 as they are not the subject matter of the invention and will not be elaborated here.

FIG. 2 shows the details of the inter-connection circuitry between the microcontroller 23 and the CD-ROM decoder 21, as well as the non-volatile memory 24 of an example of a typical IDE CD-ROM drive. As is familiar to persons skilled in the art, the microcontroller 23 of the CD-ROM drive may proceed with its normal data access operation in the drive unit under control of the host computer system. This is done by the host processor executing the CD-ROM driver software and communicating with the drive microcontroller 23 via the set of standard IDE signals in the IDE interface bus 10. This set of bus interface signals include the device addressing signals HA0-2, the input/output port selecting signals HCS0-1, the 16-bit data path HD0-15, as well as other controlling signals including the read/write strobe signal HRD/HWR, the interrupt request/acknowledge signal HIRQ/HDRAQ/HDACK, the reset signal RESET, and the 16-bit I/O transfer status indication signal IOCS16.

The apparatus for upgrading firmware code of optical disk drive via the ATA/IDE interface of the present invention, in one preferred embodiment applicable to a CD-ROM drive, allows for constructing a control circuitry such as the one illustrated in FIG. 3. This is a control circuitry based on the conventional one such as that of FIG. 1.

The basic concept of the inventive apparatus for upgrading firmware code of optical disk drives via the ATA/IDE interface is to decode the instruction issued by the host computer via the ATA/IDE interface. The decoded result determines that either the host computer system desires to maintain the normal CD-ROM drive data access operation, or it demands for an on-site upgrade of the drive firmware code via writing into the code-storage memory. When the host computer system demands an on-site firmware upgrade, the ATA/IDE interface of the CD-ROM drive is then connected to the memory device, so that the host computer may implement the writing of the new code to the memory device.

When, on the other hand, the host system requires that the CD-ROM drive unit returns back to normal operating mode, the microcontroller of the drive unit is then re-connected to the memory device, restoring its normal access of program codes in the memory.

In a preferred embodiment, upgrade of the firmware for a CD-ROM drive via the ATA/IDE interface employs to insert a controlled signal multiplexer and its related control logic between the microcontroller and its firmware storage memory device. This multiplexer is used to provide one of two possible data access routes toward the firmware memory device of the drive. One of the routes is established under normal operating condition of the optical disk drive, in which the multiplexer connects the memory device to the microcontroller so that the microcontroller can read firmware program code and data from the memory for execution. Obviously, this mode of memory access operation performed by the microcontroller on the memory device is one-directional, namely, the microcontroller performs read accesses in the memory device only.

The other route that can be established to the firmware code storage memory device is for the memory device itself to be directly accessible by the host computer system when a code upgrade is ordered. In this mode of operation, the multiplexer connects the memory device to the IDE interface of the optical disk drive unit, so that the processor of the host system can directly access the memory space of the firmware memory. This direct access includes both writing into and reading from the memory device. The ability to read from the memory device is necessary as it is a means to verify whether or not the correct content of programming has been achieved.

The on-site upgrading operation referred to herein is defined as the upgrading operation performed in the firmware code storage device of the optical disk drive without having to remove the memory device itself from inside the drive. No dedicated programming tool is needed for the upgrade operation.

Thus, as is illustrated in FIG. 3, the apparatus of the invention for upgrading the firmware code of optical disk drive via the ATA/IDE interface has a multiplexer (MUX) 36 arranged between the microcontroller 33 and the firmware code memory 34 of the drive control electronic circuitry 30. In addition to the decode processing performed by the CD-ROM decoder 31, a programming controller (PRG CNTL) 32 is used to steer the multiplexer 36 for selectively connecting the memory device 34 either to the microcontroller 33 for normal mode of operation or to the IDE bus 10 for connection with the host processor in the firmware upgrade mode of operation. In order to achieve this, as is shown in FIG. 3, the multiplexer 36 is basically a two-to-one multiplexing device, whose details will be covered below.

Therefore, when the programming controller 32 connects the firmware memory device 34 and the microcontroller 33 of the drive, both will be operating in the manner known in the art, with the microcontroller 33 performing a normal operation of accessing firmware program codes from the memory 34, as is in the case of the circuitry configuration of FIG. 1. In other words, as the programming controller 32 connects the memory device 34 to the drive microcontroller 33, the microcontroller 33 may perform its read-only access for the retrieval of its execution code from the memory device 34.

On the other hand, when the programming controller 32 connects the memory device 34 to the IDE bus 10 of the optical disk drive unit, the processor in the host computer system is then allowed to implement its access to the memory device 34. Both write and read accesses to the memory device 34 are allowed for facilitating the on-site firmware code upgrade in a series of read/write operations in the memory device. This is an operation similar to the on-board upgrading of the system BIOS (Basic Input/Output System) code for the IBM-compatible computer systems. In such on-board upgrading operations, the memory device where the BIOS code resides needs not be removed from the motherboard of the computer system, and no dedicated semiconductor memory programmer equipment is

required. With the proper setting of a few switches, the code upgrade operation can be completed by the computer system executing a software program.

Refer to FIG. 4. FIG. 4 is a schematic diagram that shows an embodiment of the controller circuitry for the CD-ROM drive device employing the apparatus of the invention for implementing on-site upgrade operation of the firmware code. In the circuitry of FIG. 4, control can be performed to selectively connect the memory device 34 either to the microcontroller 33 or to the IDE bus of the drive, in order to perform normal operation or upgrade operation respectively.

From the perspective of the controller, the circuitry of the programming controller 32 as outlined in FIG. 4 allows to convert the instruction and data issued by the host computer system via the IDE interface 10 into the electrical signals required by the memory device 34 to implement its memory content programming, as the upgrade of firmware code is ordered. Meanwhile, this circuitry of the programming controller 32 also allows for the normal connection of the microcontroller 33 and the memory device 34 as the drive normal operation is required.

In order to accomplish the conversion of the code-upgrading instructions and data, the programming controller 32 as outlined in the circuitry of FIG. 4 includes an I/O address decoder (I/O ADDR DEC) 321, an upgrade initialization key enable logic (INIT KEY EN) 322, an address program enable latch (APEN) 323, a data program enable latch (DPEN) 324, and a control program enable latch (CPEN) 325.

When an on-site upgrade operation is requested, the host computer system must issue a series of corresponding instructions and data into the IDE bus 10 in order for the programming controller 32 to be operating independently from the bus 10 itself. These instructions and data are never present over the IDE bus 10 when the drive is in the normal mode of operation. Instead, they only appear as an on-site upgrading is ordered, and these instructions and data are used to initiate the programming controller 32.

For instance, in a preferred embodiment employed in the case of an IBM-compatible system, the processor of the host system may write a series of 32 consecutive and pre-defined bytes of information at the I/O address 1F5. As is seen in FIG. 4, these data bytes are received over the bus 10A via the I/O address 1F5 as decoded by the I/O address decoder 321. The data received over I/O address 1F5 are then sent to the upgrade initialization key enable logic 322. Note that bus 10A is an extension of the IDE bus 10 that is internal to the controller circuitry 30.

The upgrade initialization key enable logic 322 is triggered by the logically positive signal at the I/O address 1F5 to receive the 32 consecutive information bytes. If these information bytes were compared and found to be compatible with a predetermined set of data, then the upgrade initialization key enable logic 322 generates an upgrade initialization key signal IKEYOK. This signal IKEYOK is then relayed to the address program enable latch 323, the data program enable latch 324, and the control program enable latch 325 respectively. These three latches are then enabled simultaneously inside the controller circuitry 30.

On the other hand, the address program enable latch 323 latches the 16-bit IDE data signals HD0-15 over the bus 10A onto the bus 3236. The counterpart 16-bit data signals on bus 3236 are designated as signals 1A0-15. Meanwhile, the data program enable latch 324 also latches the least significant byte (eight bits) of the same 16-bit IDE data signals HD0-15 over the bus 10A onto the bus 3236 as the eight-bit data signals IHD0-7. In a similar manner, the control program enable latch 325 latches the eight least significant bits of the double-byte IDE data signals HD0-15 over the bus 10A onto the bus 3236. Selected ones of the latched bits are used as the write enable signal IWE, the chip enable signal ICE, and the output enable signal IOE, as well as the IDE programming enable signal IDEPEN that are required to strobe the memory device 34 for the implementation of the programming operation.

For example, as the IDE programming enable signal IDEPEN obtains a logically positive status, the processor of the host computer system may then issue the instruction code and data necessary to program the memory device 34. These instructions and data are written into the memory device 34 over the IDE bus 10 at the designated I/O addresses. For example, in the embodiment of FIG. 4, the address of read/write access to the memory device 34 may be issued at the I/O address 1F0 over the IDE interface in data signals HD0-15. On the other hand, I/O address 1F3 and data bits HD0-7 may be used to relay the program code that is to be written into the memory device 34 at the designated address space. Further, I/O address 1F4 and data bits HD0-7 can be used to relay the necessary device strobe signals OE, CE and WE that are required by the memory device 34 when programmed. The programming enable signal IDEPEN, which is used to indicate the status of the entire CD-ROM drive concerning either the drive is in a mode of normal operation or of upgrade programming operation, can itself be relayed also via the I/O address 1F4 as a designated bit in the data by HD0-7.

As persons skilled in the art should be well aware, 16-bit address is sufficient for addressing the firmware memory device normally found in CD-ROM drives. For example, in the case of an eight-bit memory device, 16-bit address relayed via the IA0-15 signal lines are capable of addressing a memory space of 64 K bytes. For the consideration of cost reduction, eight-bit memory devices are normally used as the firmware storage device. This is suitable since optical disk drives such as CD-ROM drives are relatively slow peripheral devices when compared to the host processor of modern high-performance personal computer systems. The use of eight-bit data path as relayed via the IHD0-7 signal lines connected to the microcontroller 33 (FIG. 3) is generally adequate design arrangement for the normal CD-ROM drives. Further, all three memory device controlling enable signals IOE, IWE and ICE as well as the programming enable signal IDEPEN may all be obtained via the HD0-7 data bits relayed over the IDE interface bus.



Then, with reference to FIG. 3 again, it can be seen that the circuitry of the programming controller 32 may transfer the 16 address signals IA0-15, the eight data signals IHD0-7, and the four control signals ICE, IOE, IWE and IDEPEN onto bus 3236 at its output end. As these signals are transferred onto bus 3236, they are effectively input to the multiplexer 36, which, depending on the status of the logically positive programming enable signal IDEPEN, may then facilitate the multiplexing operation, in order to selectively connect a set of access signals onto the memory device 34. Under proper timing control, this set of relayed signals may either be used by the microcontroller 33 of the CD-ROM drive itself to operate the drive normally in the CD-ROM data reading sessions, or be used by the processor of the host computer system in the firmware upgrade session. Naturally, the semiconductor memory device 34 used in this case as the firmware storage in the CD-ROM drive must be one that is electrically reprogrammable. It can, for example, be an EEPROM.

The above descriptive paragraphs have concentrated on the operation of the electronics of a CD-ROM drive constructed in accordance with the teachings of the invention as it undergoes a firmware code upgrade operation in the memory device. On the other hand, it should be noted that, under normal situations, firmware code upgrade is a rare operation that can be expected less than several times in the lifetime of a CD-ROM drive. For most of the times, the CD-ROM drive will be reading the memory contents of the CD-ROMs inserted into the drive. In this case, the programming enable signal IDEPEN in the programming controller 32 of FIG. 4 maintains the connection of the memory device 34 substantially direct to the microcontroller 33, so that the microcontroller 33 may execute the code in the memory device 34 to implement the CD-ROM drive functionality. In this sense, a CD-ROM drive employing the apparatus of the invention is substantially the same as the conventional drives.

Thus, as is illustrated in the block diagram of FIG. 3, the host computer system can implement its firmware-upgrade code-writing operation against the

memory device directly via the standard IDE bus 10. The programming operation is executed by the host computer issuing instruction and related data via the bus 10A, and then via bus 3236 under control of the programming controller 32. The updated code is then sent over the memory device 34 via the bus 3436 as switched by the multiplexer 36. FIG. 5 details the circuitry configuration and the operation of a preferred embodiment of the multiplexer 36.

As is illustrated in FIG. 5, the multiplexer 36 that is employed in the block diagram of FIG. 3 has a group of five two-to-one multiplexer units 361, 362, 363, 364 and 365. All five multiplexer units can be switched by the same control signal. In the case of the embodiment outlined in FIGS. 3 and 4, the programming enable signal IDEPEN is used as the control signal. All the five multiplexer units has their respective first set of input tied to bus 3336 and then to the microcontroller 33 of the CD-ROM drive. The second set of input of the five multiplexer units are tied to bus 3236 and then to the IDE bus 10 via connection through the programming controller 32. The multiplexed output of the five multiplexer units are tied to bus 3436 and thus to the memory device 34. As is clearly outlined in the block diagram of FIG. 3, this bus connection arrangement allows the programming enable signal IDEPEN to facilitate the control of the memory device 34, which is to be selectively connected either to the microcontroller 33 for normal operation and to the drive IDE bus 10 for firmware upgrade operation.

Note that the multiplexer unit 361 has a switched signal path width of 16 bits to accommodate the need to relay the 16-bit IA0-15 signals that provide the address bits during the programming operation of the memory device 34. When the CD-ROM drive is operated normally, this 16-bit signal path width allows the connection of the 16-bit microcontroller address bits  $\mu$ CA0-15 to be relayed to the 16 address bits FA0-15 of the memory device 34. In a similar manner, the multiplexer unit 362 has an eight-bit width of switched signal path so as to accommodate the need to relay the two sets of data lines IHD0-7 and  $\mu$ CD0-7 onto the corresponding eight-bit data bus FD0-7 of the memory device 34. By contrast,

the multiplexer units 363, 364 and 365 are single-line multiplexing circuits that can be used to relay the WE, CE and OE control signals of the memory device 34.

Thus, with the ATA/IDE interface, a CD-ROM drive employing the apparatus for on-side firmware code upgrade may be operated in one of two operation modes as the need arises. In the normal mode of operation, the firmware code memory device and the microcontroller of the CD-ROM drive can be connected together in order that the code in the memory can be accessed by the microcontroller to conduct normal CD-ROM operation. On the other hand, when the CD-ROM drive is put in the firmware upgrade mode, the memory device can be tied to the IDE interface of the drive, allowing the processor of the host computer to write directly into the addressed memory space of the memory device via the computer system bus that is tied to the IDE bus. The programming instruction code and the new code contents can be relayed to the memory device via the IDE interface equipped in the CD-ROM drive device.

With this direct on-side programming capability, the CD-ROM drive employing the circuitry apparatus of the invention may enjoy the great convenience of firmware code on-site upgrade operation. The drive itself need not be removed from inside the computer system cabinet. The entire code-upgrading operation can be conducted fully via software control. With proper information provided, even the general end user may be able to conduct the upgrade operation by following detailed instructions shown, for example, on the display screen of the computer system. For the CD-ROM drive manufacturers, since both the upgrade software as well as the new version of the firmware program code may be made available to any requesting user via regular or electronic mail services, or via file transfer services in the public networks such as the Internet. Effective conservation of costs in both resource and time for both the manufacturer and the end user can be expected.

Further, since the apparatus for implementing on-site firmware code upgrade via the ATA/IDE interface for CD-ROM drives are relatively simple in the

nature of the digital electronic circuitry employed, they can thus be integrated easily in the ASIC chipset of the drive circuitry. In fact, as is appreciable for persons skilled in the art, the inventive apparatus may, for example, be integrated in the CD-ROM decoder 31 of FIG. 3, or even in the microcontroller itself.

While the invention has been described by way of example and in terms of preferred embodiment, it is to be understood that the invention need not be limited to the disclosed embodiments. On the contrary, it is intended to cover various modifications and similar arrangements included within the spirit of the appended claims, the scope of which should be accorded the broadest interpretation so as to encompass all such modifications and similar structures.

#### 4 . B r i e f   E x p l a n a t i o n   o f   t h e   D r a w i n g s

Other objects, features, and advantages of the present invention will become apparent by way of the following detailed description of the preferred but non-limiting embodiments. The description is made with reference to the accompanying drawings in which:

FIG. 1 is a block diagram showing the electronic controller circuitry of a conventional CD-ROM drive;

FIG. 2 shows the conventional interfacing connection between the microcontroller, the firmware memory and the CD-ROM drive decoder with respect to the drive IDE interface that is typical in the CD-ROM drive of FIG. 1;

FIG. 3 shows the block diagram of the electronic controller circuitry of the CD-ROM drive employing the apparatus for on-site firmware code upgrade in accordance with a preferred embodiment of the invention;

FIG. 4 shows the schematic diagram of the programming controller of the CD-ROM drive embodying the electronic circuit configuration of FIG. 3; and

FIG. 5 shows the schematic diagram of the multiplexer of CD-ROM drive embodying the electronic circuit configuration of FIG. 3.

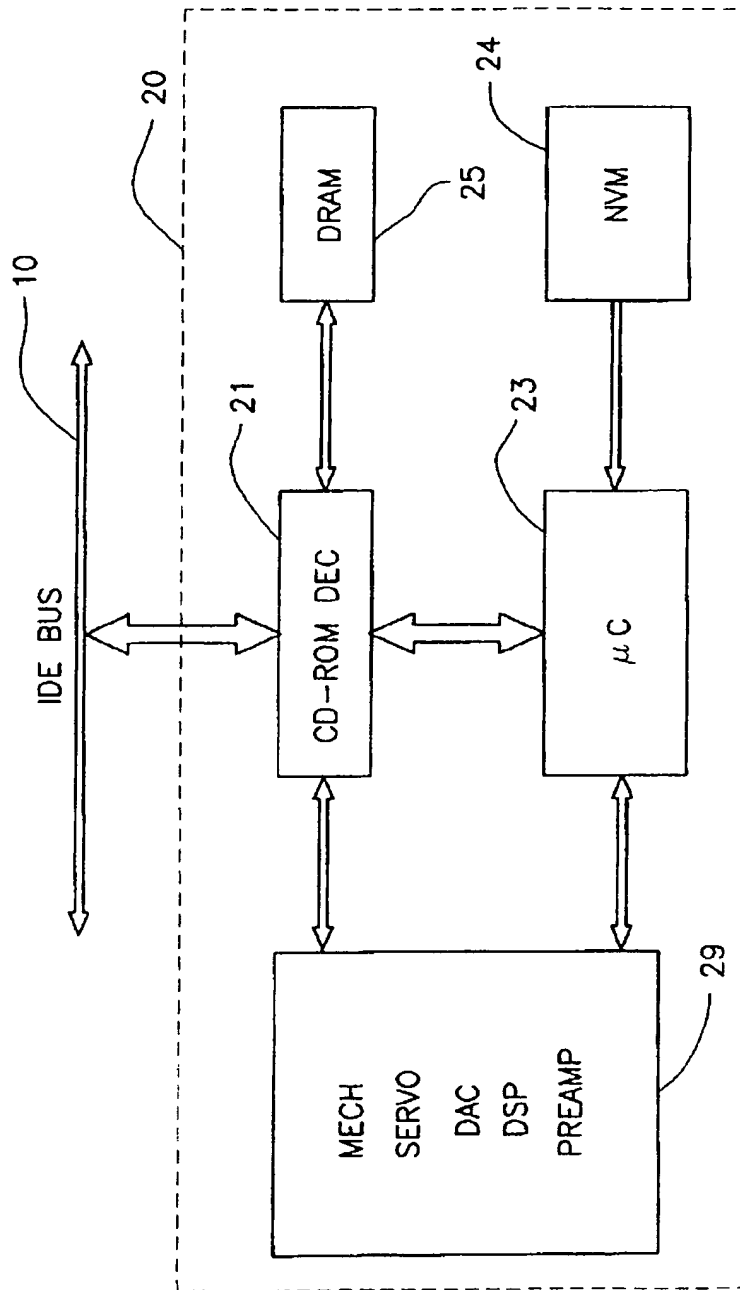


FIG. 1 (PRIOR ART)

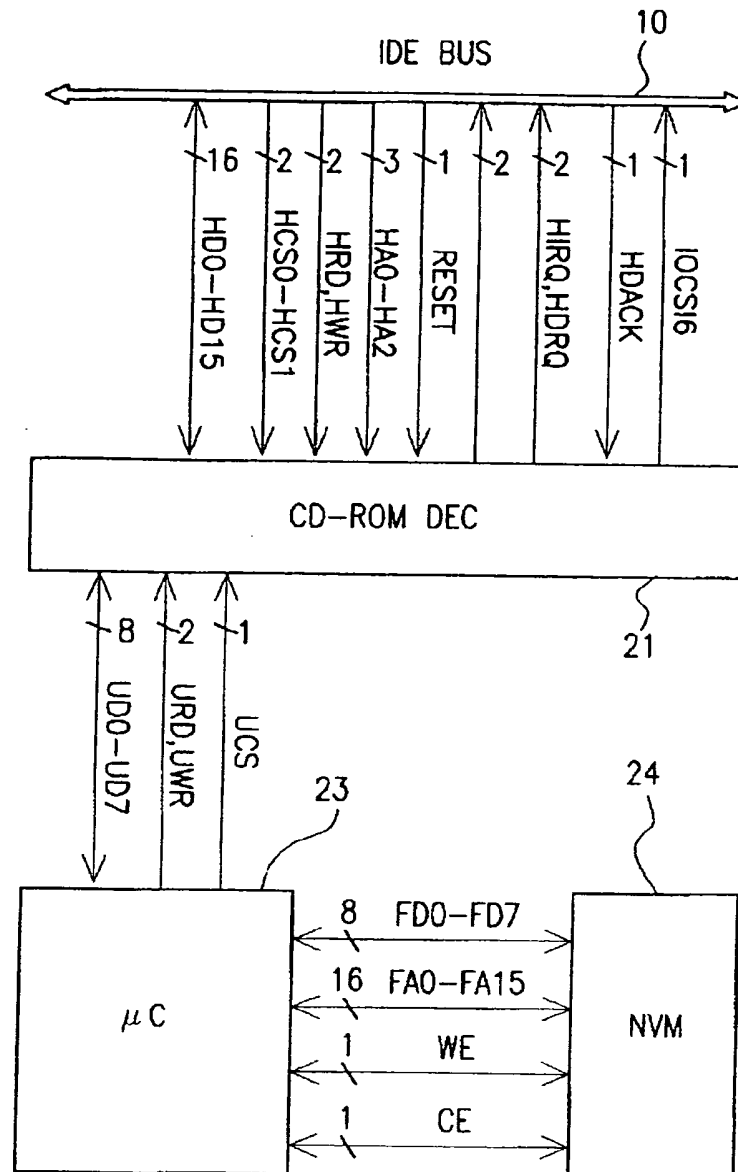


FIG. 2 (PRIOR ART)

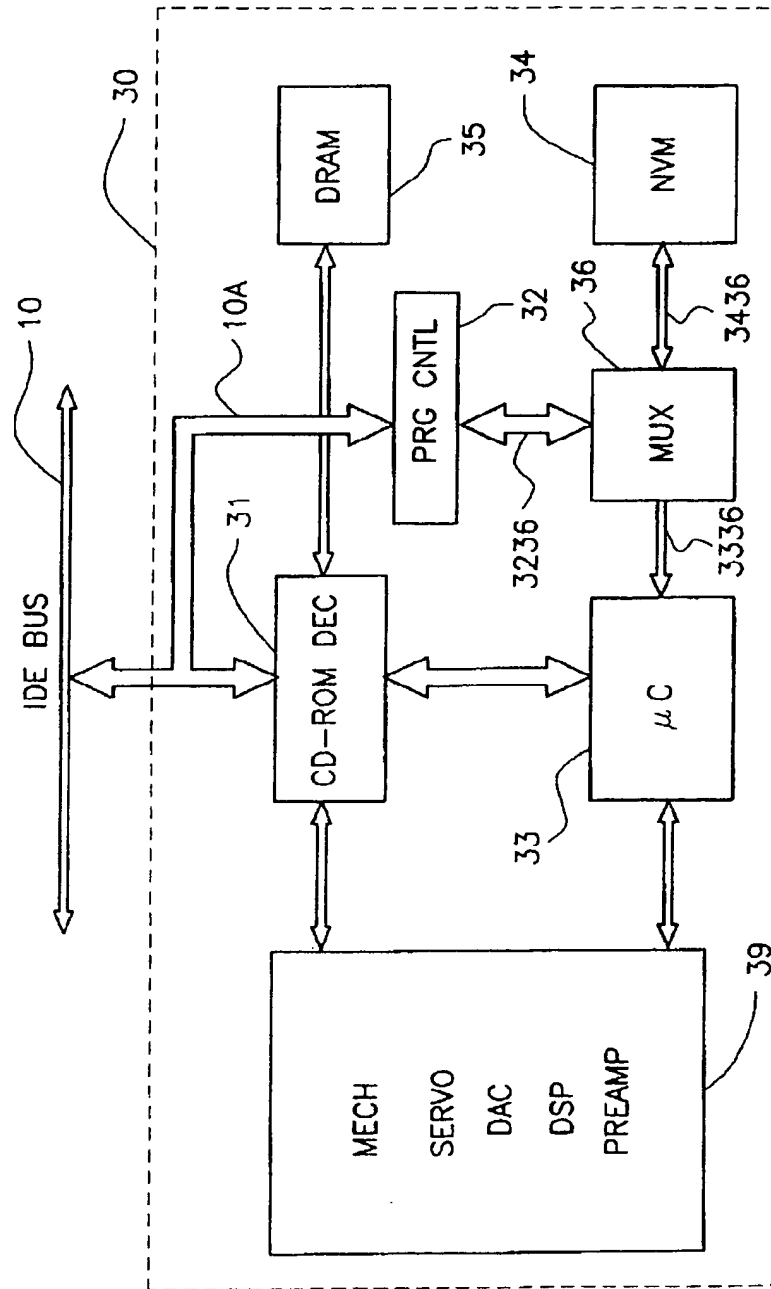


FIG. 3

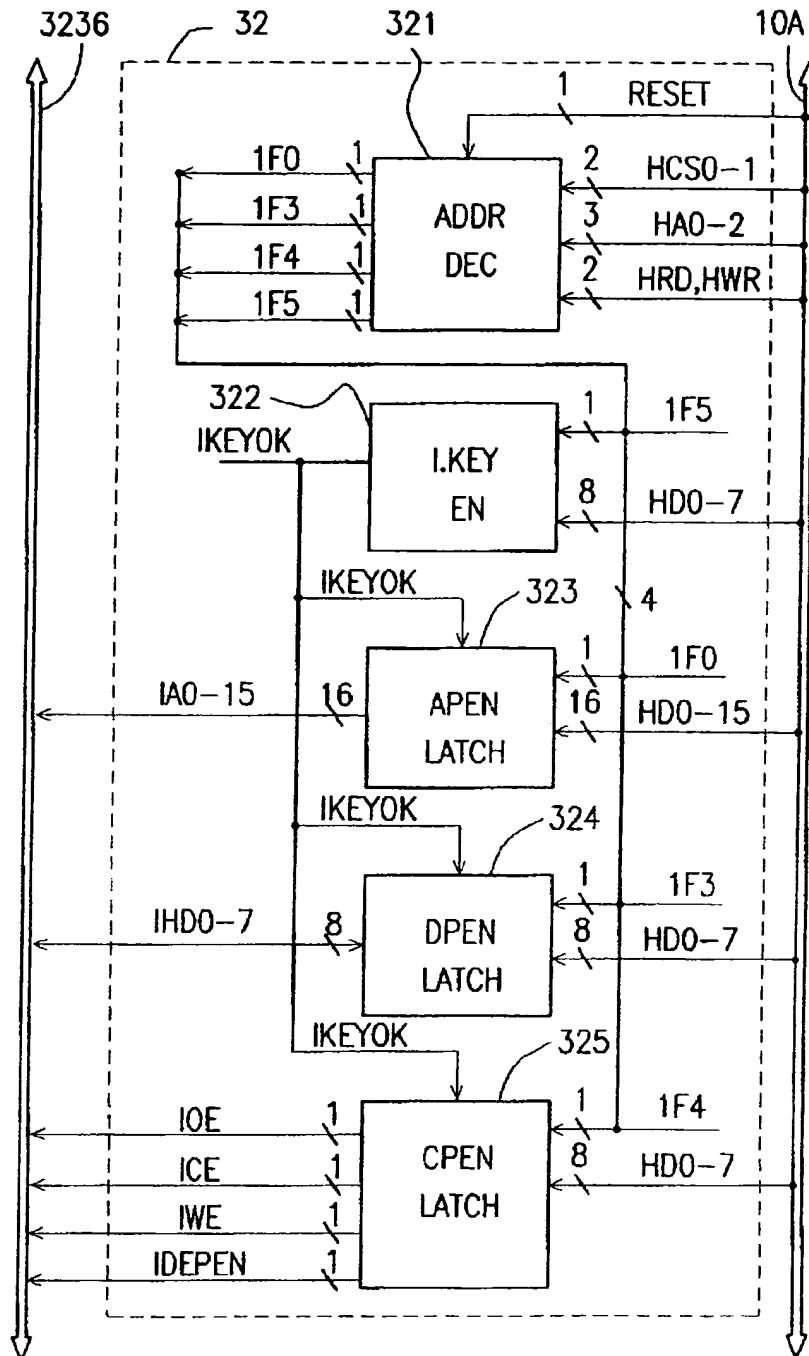


FIG. 4



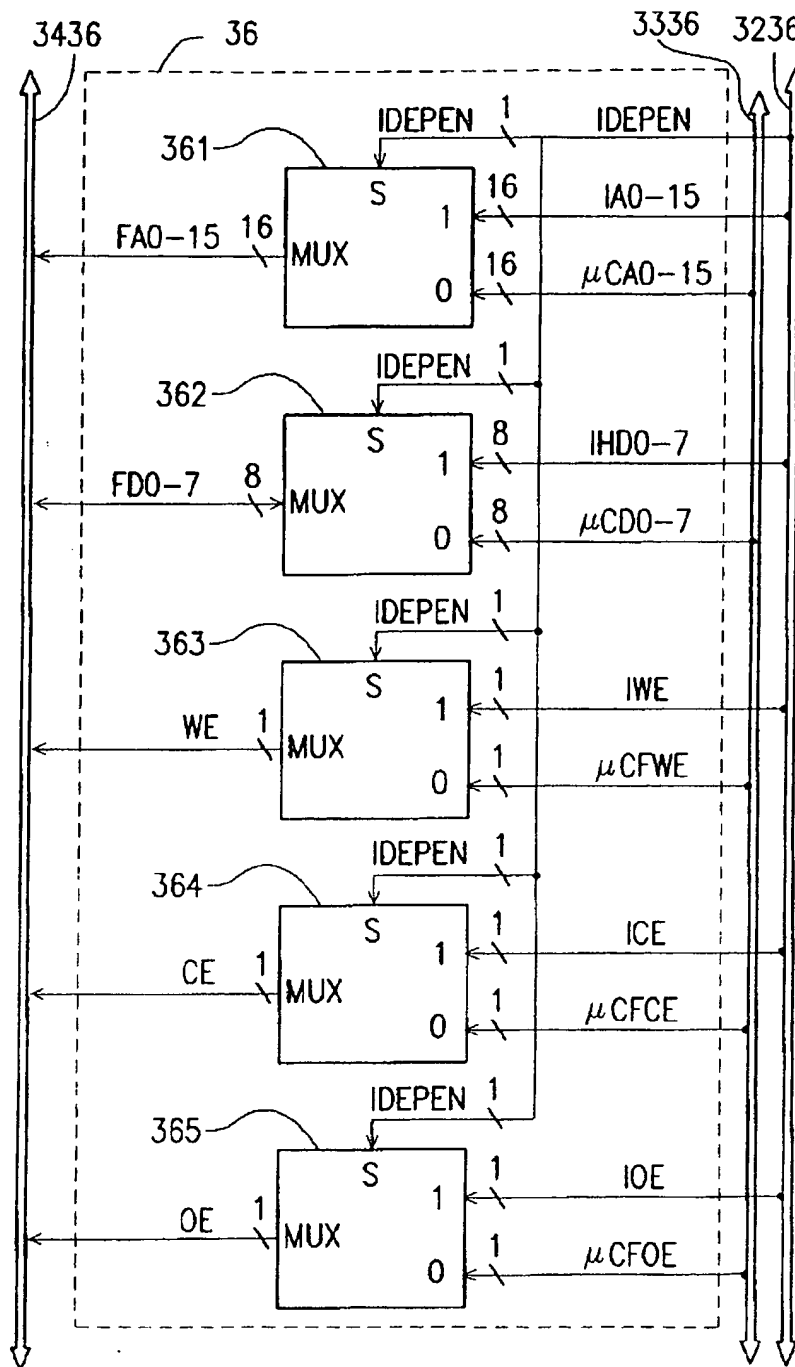


FIG. 5

1. Abstract

An apparatus and the method for upgrading the firmware code of an optical disk drive via the ATA/IDE interface are disclosed. The optical disk drive has a firmware code memory and a microcontroller that executes the firmware code. A drive decoder decodes to connect the optical disk drive to a host computer system via the ATA/IDE interface. The apparatus includes a programming controller that receives signals from the ATA/IDE interface and performs input/output decode determines whether the host computer system has requested for upgrade of the firmware or for maintaining the optical disk drive in normal operation. A multiplexer has a first input connected to the programming controller, and the second input connected to the microcontroller. The multiplexer switches the first input to the memory device via the multiplexed output for performing a firmware upgrade operation when the host computer system requests for a firmware upgrade. The on-site upgrade operation can be performed without access to the inside of the drive unit.

2 . R e p r e s e n t a t i v e   D r a w i n g

F i g . 3